

WHAT IS CLAIMED IS:

1. A data access method, comprising a data reading procedure to read a certain bit range of data from a data storage zone, said certain bit range being stored in said data storage zone from a starting bit address (a) to an end bit address (b), and said data reading procedure comprising steps of:

 performing a first operation of said starting bit address (a) to obtain a first shift S1;

 performing a second operation of said starting bit address (a) to obtain a second shift S2;

 performing a first shift operation of said data with said first shift S1 to obtain a first shifted data unit;

 performing a second shift operation of said data with said second shift S2 to obtain a second shifted data unit; and

 synthesizing said first and said second shifted data units to obtain a read data unit.

2. The data access method according to claim 1 wherein said data storage zone stores data as at least one data unit consisting of m bits, and said bit range consists of n bits, where n is greater than m.

3. The data access method according to claim 2 wherein said first and said second operations are performed by the following formulae:

$$S1 = \text{mod } [a, m]; \text{ and}$$

$$S2 = m - \text{mod } [a, m] = m - S1,$$

where $\text{mod } [a, m]$ is the remainder on division of a by m.

4. The data access method according to claim 3 wherein said first shift operation is performed by shifting a first data unit of said data to be read toward one of the higher bit direction and the lower bit direction, and said

second shift operation is performed by shifting a second data unit of said data to be read toward the other of the higher bit direction and the lower bit direction.

5. The data access method according to claim 4 wherein said second data unit is immediately adjacent to said first data unit in said data storage zone.
6. The data access method according to claim 5 wherein said first and said second shift operations are further performed on subsequent data units until a data unit comprising said end data bit address (b) has been shifted to obtain a last shifted data unit.
7. The data access method according to claim 6 further comprising a step of masking said last shifted data unit with a mask data MD for clearing bits excluded from said bit range, where $MD = 0xFF \gg (m - (b-a+1))$, the expression “0xFF” indicates an 8-bit hexadecimal mask data and the 8 bits are all “1”, and the expression “X >> Y” indicates the rightward shift of the data X by Y bits.
8. The data access method according to claim 1 wherein said first and said second shifted data units are synthesized via an OR gate operation.
9. A data access method, comprising a data writing procedure to write a certain bit range of data into a data storage zone, said certain bit range being stored into said data storage zone from a starting bit address (a) to an end bit address (b), and said data writing procedure comprising steps of:
 - performing a first operation of said starting bit address (a) to obtain a first shift S3;
 - performing a second operation of said starting bit address (a) to obtain a second shift S4;
 - performing a first shift operation of said data with said first shift S3 to

obtain a first shifted data unit;

performing a second shift operation of said data with said second shift S4 to obtain a second shifted data unit; and

synthesizing said first and said second shifted data units to obtain a written data unit.

10. The data access method according to claim 9 wherein said data storage zone stores data as at least one data unit consisting of m bits, and said bit range consists of n bits, where n is greater than m.

11. The data access method according to claim 10 wherein said first and said second operations are performed by the following formulae:

$$S3 = \text{mod } [a, m]; \text{ and}$$

$$S4 = m - \text{mod } [a, m] = m - S3,$$

where $\text{mod } [a, m]$ is the remainder on division of a by m.

12. The data access method according to claim 11 wherein said first shift operation is performed by shifting a first data unit of said data to be written toward one of the higher bit direction and the lower bit direction, and said second shift operation is performed by shifting a second data unit of said data to be written toward the other of the higher bit direction and the lower bit direction.

13. The data access method according to claim 12 wherein said second data unit is immediately adjacent to said first data unit in said data storage zone.

14. The data access method according to claim 13 further comprising before said first and said shifting operations steps of:

determining whether said second data unit is the last data unit of said data to be written; and

masking said second data unit with a mask data MD3 for clearing bits

excluded from said bit range when said second data unit is the last data unit of said data to be written, where $MD3 = 0xFF \ll (\text{mod } [b, m] + 1)$, $\text{mod } [b, m]$ is the remainder on division of b by m, the expression “0xFF” indicates an 8-bit hexadecimal mask data and the 8 bits are all “1”, and the expression “ $X \ll Y$ ” indicates the leftward shift of the data X by Y bits.

15. The data access method according to claim 13 wherein said first and said second shift operations are further performed on subsequent data units until the last data unit of said data to be written has been shifted.

16. The data access method according to claim 10 further comprising steps of:
performing a third shifting operation of a starting data unit of said data to be written with said first shift S3; and

masking said starting data unit with a mask data MD2 for clearing bits excluded from said bit range,

where $MD2 = \sim(0xFF \ll S3)$, the expression “0xFF” indicates an 8-bit hexadecimal mask data and the 8 bits are all “1”, the expression “ $X \ll Y$ ” indicates the leftward shift of the data X by Y bits, and the expression “ $\sim Z$ ” indicates the reverse logic operation of data Z.

17. The data access method according to claim 9 wherein said first and said second shifted data units are synthesized via an OR gate operation.

18. A data access method, comprising a data writing procedure to write a certain bit range of data into a data storage zone, said data storage zone storing data as at least one data unit consisting of m bits, said certain bit range consisting of n bits and being stored into said data storage zone from a starting bit address (a) to an end bit address (b), and said data writing procedure comprising steps of:

performing a first operation of said starting bit address (a) and said bit

number m to obtain a first shift S3;

performing a second operation of said starting bit address (a) and said bit number m to obtain a second shift S4;

performing a first clear and writing procedure of said data to be written when n is no greater than m, said first clear and writing procedure comprising a step of masking said data to be written with a first mask data $MD1 = \sim((0xFF >> ((m-1) - b + a)) << S3)$;

performing a second clear and writing procedure of said data to be written when n is greater than m, said second clear and writing procedure comprising a step of masking the starting data unit of said data to be written with a second mask data $MD2 = \sim(0xFF << S3)$;

performing a third clear and writing procedure of said data to be written when n is greater than m, said third clear and writing procedure comprising a step of masking the end data unit of said data to be written with a third mask data $MD3 = 0xFF << (\text{mod } [b, m] + 1)$; and

performing a first and a second shift operations of said data with said first and said second shifts S3 and S4 to obtain a first and a second shifted data units, and synthesizing said first and said second shifted data units to obtain a written data unit when n is greater than m;

where the expression “0xFF” indicates a hexadecimal mask data, the expression “ $X >> Y$ ” indicates the rightward shift of the data X by Y bits, the expression “ $X << Y$ ” indicates the leftward shift of the data X by Y bits, the expression “ $\sim Z$ ” indicates the reverse logic operation of data Z, the expression “ $X \& Y$ ” indicates AND gate operation of data X and Y, and the expression “ $\text{mod } [b, m]$ ” indicates the remainder on division of b by m.

19. The data access method according to claim 18 wherein said data writing

procedure is performed as little endian.

20. The data access method according to claim 18 wherein said data writing procedure is performed as big endian.